Lógica en la Informática / Logic in Computer Science

SOLUTIONS. Tuesday November 15, 2011

Time: 1h45min. No books, lecture notes or formula sheets allowed.

1) We already know the *ladder encoding* for *at-most-one*, that is, for expressing in CNF that at most one of $x_1 \dots x_n$ is true. It uses n auxiliary variables a_i, where each a_i means "at least one of $x_1 \dots x_i$ is true". We also know that an encoding for at-most-one is *arc-consistent for unit propagation* if, whenever one of $x_1 \dots x_n$ becomes true, the SAT solver's unit propagation mechanism will set the other variables to false.

1a List, without giving any explanations, which and how many clauses are used in this encoding. Argue whether the encodng is arc-consistent for unit propagation or not.

`Solution:` For each i in $1..n$ there will be three clauses:

A) $\neg x_i \vee a_i$
B) $\neg a_i \vee a_{i+1}$ (this clause only for i in $1 \dots n-1$)
C) $\neg a_i \vee \neg x_{i+1}$ (this clause only for i in $1 \dots n-1$).

Total: $3n-2$ clauses. To prove that it is arc-consistent for unit propagation, assume some x_i (with i in $1..n$) becomes true. Then by successive unit propagations, the following literals become true too:

To the right:

a_i	by A) $\neg x_i \vee a_i$
a_{i+1}	by B) $\neg a_i \vee a_{i+1}$
a_{i+2}	by B) $\neg a_{i+1} \vee a_{i+2}$
$\vdots$	$\vdots$
a_n	by B) $\neg a_{n-1} \vee a_n$
$\neg x_{i+1}$	by C) $\neg a_i \vee \neg x_{i+1}$
$\vdots$	$\vdots$
$\neg x_n$	by C) $\neg a_{n-1} \vee \neg x_n$

and to the left:

$\neg a_{i-1}$	by C) $\neg a_{i-1} \vee \neg x_i$
$\neg a_{i-2}$	by B) $\neg a_{i-2} \vee a_{i-1}$
$\neg a_{i-3}$	by B) $\neg a_{i-3} \vee a_{i-2}$
$\vdots$	$\vdots$
$\neg a_1$	by B) $\neg a_1 \vee a_2$
$\neg x_{i-1}$	by A) $\neg x_{i-1} \vee a_{i-1}$
$\vdots$	$\vdots$
$\neg x_1$	by A) $\neg x_1 \vee a_1$

1b Extend the ladder encoding for *at-most-two* constraints, with variables a_i and b_i, where b_i means "at least two of $x_1 \dots x_i$ are true". List, without giving any explanations, which and how many clauses are used, and how many auxiliary variables.

`Solution:` For each i in $1..n$ there will be five clauses:

A) $\neg x_i \vee a_i$
B) $\neg a_i \vee a_{i+1}$
C) $\neg b_i \vee b_{i+1}$
D) $\neg a_i \vee \neg x_{i+1} \vee b_{i+1}$
E) $\neg b_i \vee \neg x_{i+1}$

where types clauses B,C,D,E exist only for i in $1 \dots n-1$. Total: $5n-4$ clauses.

1c Argue whether this at-most-two encoding is arc-consistent for unit propagation or not (that is, whenever two of $x_1 \dots x_n$ become true, unit propagation sets the other variables to false).

`Solution:` To prove that it is arc-consistent for unit propagation, assume some x_i and x_j with $1 \le i < j \le n$ become true. Then the successive unit propagations are:

To the right:

a_i	by A) $\neg x_i \vee a_i$
a_{i+1}	by B) $\neg a_i \vee a_{i+1}$
a_{i+2}	by B) $\neg a_{i+1} \vee a_{i+2}$
$\vdots$	$\vdots$
a_n	by B) $\neg a_{n-1} \vee a_n$
b_j	by D) $\neg a_{j-1} \vee \neg x_j \vee b_j$
b_{j+1}	by C) $\neg b_j \vee b_{j+1}$
b_{j+2}	by C) $\neg b_{j+1} \vee b_{j+2}$
$\vdots$	$\vdots$
b_n	by C) $\neg b_{n-1} \vee b_n$
$\neg x_{j+1}$	by E) $\neg b_j \vee \neg x_{j+1}$
$\vdots$	$\vdots$
$\neg x_n$	by E) $\neg b_{n-1} \vee \neg x_n$

and to the left:

$\neg b_{j-1}$	by E) $\neg b_{j-1} \vee \neg x_j$
$\neg b_{j-2}$	by C) $\neg b_{j-2} \vee b_{j-1}$
$\vdots$	$\vdots$
$\neg b_1$	by C) $\neg b_1 \vee b_2$
$\neg a_{i-1}$	by D) $\neg a_{i-1} \vee \neg x_i \vee b_i$
$\neg a_{i-2}$	by B) $\neg a_{i-2} \vee \vee a_i$
$\vdots$	$\vdots$
$\neg a_1$	by B) $\neg a_1 \vee \vee a_2$
$\neg x_{i-1}$	by A) $\neg x_{i-1} \vee a_{i-1}$
$\vdots$	$\vdots$
$\neg x_1$	by A) $\neg x_1 \vee a_1$

Finally, for each x_k with $i < k < j$ (if there is any), the literals a_{k-1} and $\neg b_k$ have been propagated, and hence by D) $\neg a_{k-1} \vee \neg x_k \vee b_k$, these literals $\neg x_k$ are also propagated. Altogether, if variables x_i and x_j with $1 \le i < j \le n$ become true, then by unit propagation all other variables in $x_1 \dots x_n$ become false, thus proving arc consistency.

2) We are given two very large propositional formulas F_1 and F_2, in a simple format. For example, $x_1 \wedge ((x_2 \vee x_3) \vee \neg(x_y \wedge x_5))$ is written like:

```
0=and(x1,1)
1=or(2,3)
2=or(x2,x3)
3=not(4)
4=and(x4,x5)
```

Explain in the simplest possible way what you would do for using the Barcelogic SAT solver to determine whether F_1 and F_2 are logically equivalent.

`Solution:` To use the Barcelogic solver (or any other state-of-the-art SAT solver) the input has to be a CNF formula (a set of clauses). We can use the Tseitin transformation to transform these formulas into CNF: we introduce a new variable for each non-leaf node of a formula. If the node is of the form $n = and(x, y)$, we generate three clauses: $\neg z_n \vee x$, $\neg z_n \vee y$, and $z_n \vee \neg x \vee \neg y$. If the node is of the form $n = or(x, y)$, we generate three clauses: $z_n \vee \neg x$, $z_n \vee \neg y$, and $\neg z_n \vee x \vee y$. If the node is of the form $n = not(x)$, we generate the two clauses $z_n \vee \neg x$, and $\neg z_n \vee x$ (or simply replace all occurrences of z_n by $\neg x$, introducing no variable z_n for this node).

Let the sets of clauses S_1 and S_2 be the resulting Tseitin transformations of F_1 and F_2, using names z^1 and z^2 respectively for the new node variables. So the root variables are z_0^1 and z_0^2.

Logical equivalence of F_1 and F_2 means that they have the same models, that is, there is no model of $S_1 \cup S_2$ such that one of z_0^1 and z_0^2 is true and the other one is false. So we can call Barcelogic with input $S_1 \cup S_2 \cup \{ \neg z_0^1 \vee \neg z_0^2, \;\; z_0^1 \vee z_0^2 \}$, and it will return unsatisfiable iff F_1 and F_2 are logically equivalent.

Note: one can also directly check unsatisfiability of $(F_1 \wedge \neg F_2) \vee (F_2 \wedge \neg F_1)$ i.e., if S is the Tseitin transformation of this formula, with root symbol z_0, Barcelogic returns "unsatifiable" on the clause set $S \cup \{ z_0 \}$ iff F_1 and F_2 are logically equivalent. But this is less efficient (think why).

3) A very large chain of supermarkets sells its products on the internet to N customers. They know, for each one of their P products, which customers have bought it. Now they want to make a survey study (ask a number of questions) to a subset of at most K of their customers, in such a way that for each one of their products, at least one of the buyers of that product is in the surveyed subset. Describe how to use a SAT solver to find a an adequate subset of K customers. (if it exists).

`Solution:` We encode the problem into a CNF such that a solution exists iff the SAT solver, when run on this CNF, returns a model. Moreover, from each model a solution can be easily reconstructed.

Let $\{1 \dots N\}$ be the set of all customers, and, for each product $p \in \{1 \dots P\}$, let B_p be the set of its buyers, with $B_p \subseteq \{1 \dots N\}$.

Solution 3A: Consider N propositional variables x_i, for each i in $\{1 \dots N\}$, meaning that "customer i is in the surveyed subset". Then we can express for each product p that at least one of its buyers is in the surveyed subset, with a single clause $\bigvee_{i \in B_p} x_i$ (for each product p, one clause with $|B_p|$ literals). In addition we need to express that at most K customers are in the surveyed subset. This we can do with a cardinality constraint $x_1 + \ldots + x_N \leq K$ that can be encoded, for instance, with a sorting network using $O(N \log^2 N)$ clauses and the same number of auxiliary variables.

Solution 3B: We can consider propositional variables $x_{i,j}$, for i in $\{1 \dots N\}$ and j in $\{1 \dots K\}$, meaning that "customer i is the j-th member of the surveyed subset". Then we can express for each product p that at least one of its buyers is one of the K members of the surveyed subset, with a single clause $\bigvee_{i \in B_p} (x_{i,1} \vee \ldots \vee x_{i,K})$. This requires one clause with $|B_p| \cdot K$ literals for each product p. In addition, for each j in $\{1 \dots K\}$, we need to express that at most one customer is the j-th member of the surveyed subset. This we can do with for each j in $\{1 \dots K\}$ an at-most-one constraint $x_{1,j} + \ldots + x_{N,j} \leq 1$, using any of the known encodings (quadratic, ladder, Heule, log, ...).